EXCEPTION MECHANISM FOR A COMPUTER

ABSTRACT

A computer has a multi-stage execution pipeline and an instruction decoder. The instruction decoder is designed (a) to decode instructions of a complex instruction set for execution by the pipeline, the instruction set being architecturally exposed for execution by user-state programs stored in a main memory of the computer, the instruction set having variable-length instructions and many instructions having multiple side-effects and a potential to raise multiple exceptions, (b) for at least some instructions of the complex instruction set, to issue two or more instructions in a second internal form into the execution pipeline; (c) to generate information descriptive of instructions to be executed by the pipeline, and to store the information into non-pipelined processor registers of the computer; and (d) to determine whether instructions will complete in the pipeline, and to abstain from writing descriptive information into the processor registers for instructions following an instruction determined not to complete. The pipeline exception circuitry is designed to recognize an exception occurring in an instruction after a first side-effect of the instruction has been architecturally committed, and thereupon, to architecturally expose in the processor registers information describing a processor state of the computer, including an intra-instruction program counter value, and to transfer execution to an exception handler. Pipeline resumption circuitry is effective, after completion of the software exception handler, to resume execution of the excepted program based on the information in the processor registers. The processor registers of the computer are designed to architecturally expose sufficient information about the state of the excepted instruction that the transfer and resume are effected without saving processor state to the main memory, the processor registers and general purpose registers of the computer together providing sufficient working storage for execution of the exception handler and resumption of the program, without storing processor state to the main memory.